

REMARKS

Claims 1-20 remain in this application for further consideration in light of the foregoing remarks. No new matter has been added.

I. Interview of December 13, 2005

Applicants' attorney held an Interview with Examiner Fowlkes on December 13, 2005. During the interview, applicants' attorney discussed claim language and the cited references with Examiner Fowlkes. An agreement was not reached during the interview. Examiner Fowlkes, however, requested that the claim suggestions be incorporated into an amended so that they may be formally considered and further searching done if required. Applicants submit this amendment in hopes of capturing the essence of the interview, clarifying the invention, and furthering prosecution.

II. Description of a Few Aspects of the Disclosure

Applicants' attorney discussed a few aspects of the disclosure during the December 13, 2005 interview. Applicants reiterate a few of those aspects herein. The portions cited herein are for explanatory purposes only and not meant to limit the claims in any manner. The Background of the application specifically recites as follows:

"As the stack is unwound, it is necessary to recover the values of preserved registers that were saved by each procedure in order to reconstruct the previous frame's context. When a procedure's frame is removed from the call stack the preserved registers for its corresponding procedure must be reloaded with its saved values of local variables. The information about which preserved register is saved for a given procedure and where it is saved (e.g., memory or another register) is generated by the compiler as unwind data, stored in the binary text segment of the program itself, according to a particular programming convention. Unwind data, sometimes known as metadata, is a description of information related to a contiguous sequence of instructions of the program.

However, after source code for a computer program has been compiled, tools may be employed to insert code for profiling, or to reorder and optimize basic blocks of the code, or to otherwise instrument the code in a manner, that perturbs the binary code. When the binary code has been perturbed, the unwind data may no longer reflect the correct information necessary for the proper execution of the program during exceptions. The traditional approach for "fixing" the unwind data is to perform the modifications at the source code level, recompile and relink the computer programs. However, such an approach is a potentially expensive and lengthy process. Moreover, returning to the source code may not provide the flexibility required." *Background*, at page 2, lines 4-21.

The Specification of the application specifically recites as follows:

Runtime 210 receives notifications from operating system 105 in FIGURE 1 that an exception has occurred. It is further programmed to unwind stack region 300 in response to such a notification. Runtime 210 can access the appropriate metadata 201 or 202 stored in program modules 206. Runtime 210 is also programmed to create state records (not shown) describing the locations of relevant variables, registers, and the like.

Unwind rewriter 208 reconstructs metadata 201-202 when binary code of procedure P1 or procedure P2 has been perturbed. Unwind rewriter 208 receives information about basic blocks (not shown) for procedures (P1 and P2) along with metadata 201-202. Unwind rewriter 208 evaluates the impact of the code perturbation upon metadata 201-202 and rewrites metadata 201-202 as though the basic blocks were unperturbed. *Specification*, at page 9, lines 9-19.

III. Rejection of Claims 1-20 Under 35 U.S.C. 102(b)

Claims 1-20 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,802,371 issued to Meier (hereinafter "Meier"). Even though applicants respectfully disagree with the rejection, claims 1, 4, 11, 15, 19 and 20 have been amended for clarification in hopes of expediting this matter.

Applicants' amended independent claim 1 specifically recites the following elements that are not taught or suggested by the Meier reference:

a runtime for generating unwind data, wherein the unwind data includes a first plurality of blocks of metadata having a first order of blocks, wherein each block of metadata is associated with a corresponding procedure in the plurality of procedures; and

an unwind rewriter programmed to obtain the unwind data and reorder the first plurality of blocks of metadata to generate a second plurality of blocks of metadata having a second order, wherein the first plurality of blocks are reordered in response to a modification of the sequence of binary instructions within a procedure, such that the second plurality of blocks of metadata accurately represents the modified sequence of binary instructions.

Applicants' amended independent claim 4 specifically recites the following elements that are not taught or suggested by the Meier reference:

obtaining original unwind data that describes the original order of the basic blocks;

regenerating new unwind data from the original unwind data, wherein the new unwind data includes *a reordering of the original order of basic blocks*, and wherein the reordering represents the current order of basic blocks within the modified binary procedure; and

writing the new unwind data to the modified binary procedure.

Applicants' amended independent claim 11 specifically recites the following elements that are not taught or suggested by the Meier reference:

"receiving unwind data comprising an unwind table and a plurality of unwind descriptor records wherein the unwind data is associated with the unmodified procedure;

parsing the unwind data to identify a start basic block and an end basic block for a region associated with the procedure; and

rewriting the unwind data, wherein the rewriting of unwind data includes a reordering of unwind data, a second unwind table and a second plurality of unwind descriptor records such that the rewritten unwind data accurately represents the binary modification to the procedure."

Applicants' amended independent claim 15 specifically recites the following elements that are not taught or suggested by the Meier reference:

A computer-readable medium encoded with a data structure for providing metadata representing a binary program including basic blocks that have been *modified from an original unwind order to an unwind rewritten order*, the data structure comprising:

a second metadata field associated with a second group of instructions within the first procedure, wherein the first metadata field and the second metadata field accurately *reflect a flow of instructions of the basic blocks in the unwind rewritten order, and wherein the flow of instructions of the basic blocks in the unwind rewritten order includes a reordering of a flow of instructions of the basic blocks in the original unwind order.*

Applicants' amended independent claim 19 specifically recites the following elements that are not taught or suggested by the Meier reference:

A computer-readable medium having computer-executable instructions for *rewriting unwind data* in response to a binary modification to a procedure, the procedure including a plurality of basic blocks, the instructions comprising:

receiving unwind data comprising an unwind table and a plurality of unwind descriptor records wherein the unwind data is associated with the unmodified procedure;

parsing the unwind data to identify a start basic block and an end basic block for a region associated with the procedure; and

rewriting the unwind data, wherein the *rewritten unwind data includes a reordering of the unwind data*, a second unwind table and a second plurality of unwind descriptor records such that the rewritten unwind data accurately represents the binary modification to the procedure.

Applicants' amended claim 20 specifically recites the following elements that are not taught or suggested by the Meier reference:

A computer-readable medium having computer-executable instructions for *rewriting unwind data* wherein a current order of basic blocks within the modified binary procedure differs from an original order of the basic blocks, the instructions comprising:

obtaining *original unwind data that describes the original order of the basic blocks*;

rewriting the original unwind data, wherein the rewritten unwind data includes a *reordering of the original order of basic blocks*, and wherein the reordering represents the current order of basic blocks within the modified binary procedure; and

writing the rewritten unwind data to the modified binary procedure.

The cited reference does not teach all of the limitations that are recited in applicants' independent claims 1, 4, 11, 15, 19 and 20. Meier addresses problems associated with debugging distributed programs. *Meier*, at col. 2, lines 34-40. As defined in Meier, a distributed program may include client-server or peer-to-peer applications. *Meier*, at col. 1, lines 34-40. Meier teaches "a method of, and system for, determining the call relations of a distributed program without the performance overhead of recording relations in a table." *Meier*, at col. 2, lines 18-23. Generally, Meier teaches that "a user may wish to debug the distributed set of client-server programs as if they were on a single program." *Meier*, at col. 2, lines 18-23. Meier continues by teaching that "[t]he present invention meets this need by providing a *distributed call stack*, wherein the call stacks of the client and server programs are *appended together to form a single distributed call stack*." *Meier*, at col. 4, line 66 - col. 5, line 2.

Column 2 of Meier is a summary that specifically recites, in pertinent part, as follows:

"For example, when a breakpoint is encountered in an RPC server program while using a debugger for distributed programs, the *call stacks* for the client and server program are *appended together into a single distributed call stack*. In the case of nested RPC calls (e.g. program A executes an RPC call to program B which executes an RPC call to C and so on) all of the *call stacks are appended together* in the order they were created. The distributed call stack may span many programs, threads of execution, and computing machines. *Meier*, at col. 2, lines 31-40. (Emphasis added).

Here, Meier is teaching walking up the call stack in a distributed setting. Meier does not teach further processing of a call stack that has already been generated. As recited in claim 1, Meier does not teach the combination of "a runtime for generating unwind data" and "an unwind rewriter programmed to obtain the unwind data and reorder the first plurality of blocks of metadata to generate a second plurality of blocks of metadata having a second order." Independent claims 4, 11, 15, 19 and 20 make a similar distinction. Accordingly, applicants


believe that independent claims 1, 4, 11, 15, 19 and 20 are clearly allowable under 35 U.S.C. §102(b).

Regarding claims 2-3, 5-10, 12-14 and 16-18, applicants assert that the limitations of those claims are not taught or otherwise suggested by the cited art. Moreover, claims 2-3, 5-10, 12-14 and 16-18 ultimately depend from independent claims 4, 11, 15, 19 and 20, respectively. Claims 4, 11, 15, 19 and 20 are clearly allowable as more fully stated above. Accordingly, applicants assert that claims 2-3, 5-10, 12-14 and 16-18 are also allowable for at least those same reasons.

In view of the foregoing, applicants respectfully request a Notice of Allowance. If the Examiner believes a telephone conference would advance the prosecution of this application, the Examiner is invited to telephone the undersigned at the below-listed telephone number.

Respectfully submitted,

MERCHANT & GOULD P.C.


Ryan T. Grace
Reg. No. 52,956
206.342.6258

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903

27488

PATENT TRADEMARK OFFICE